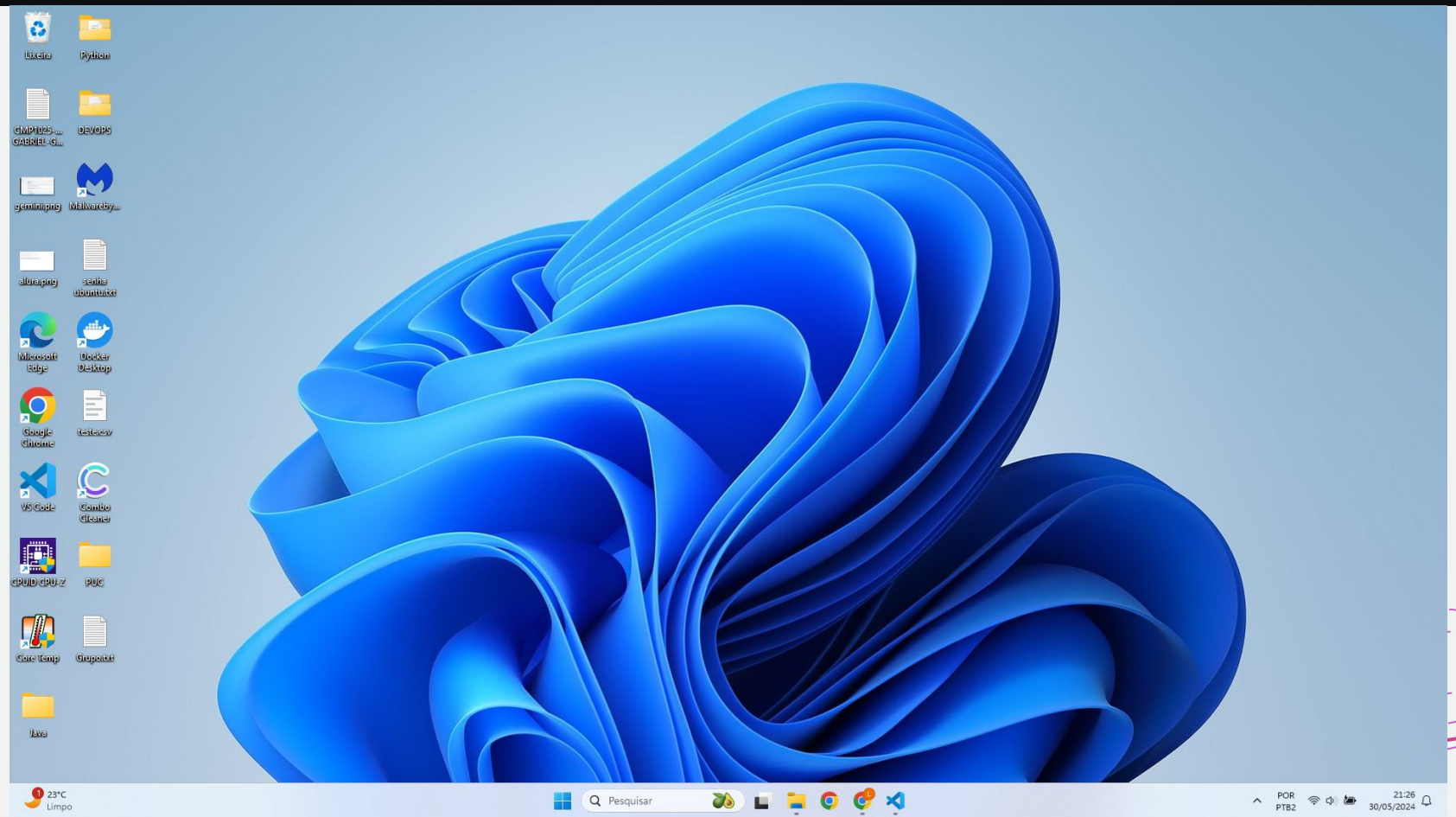
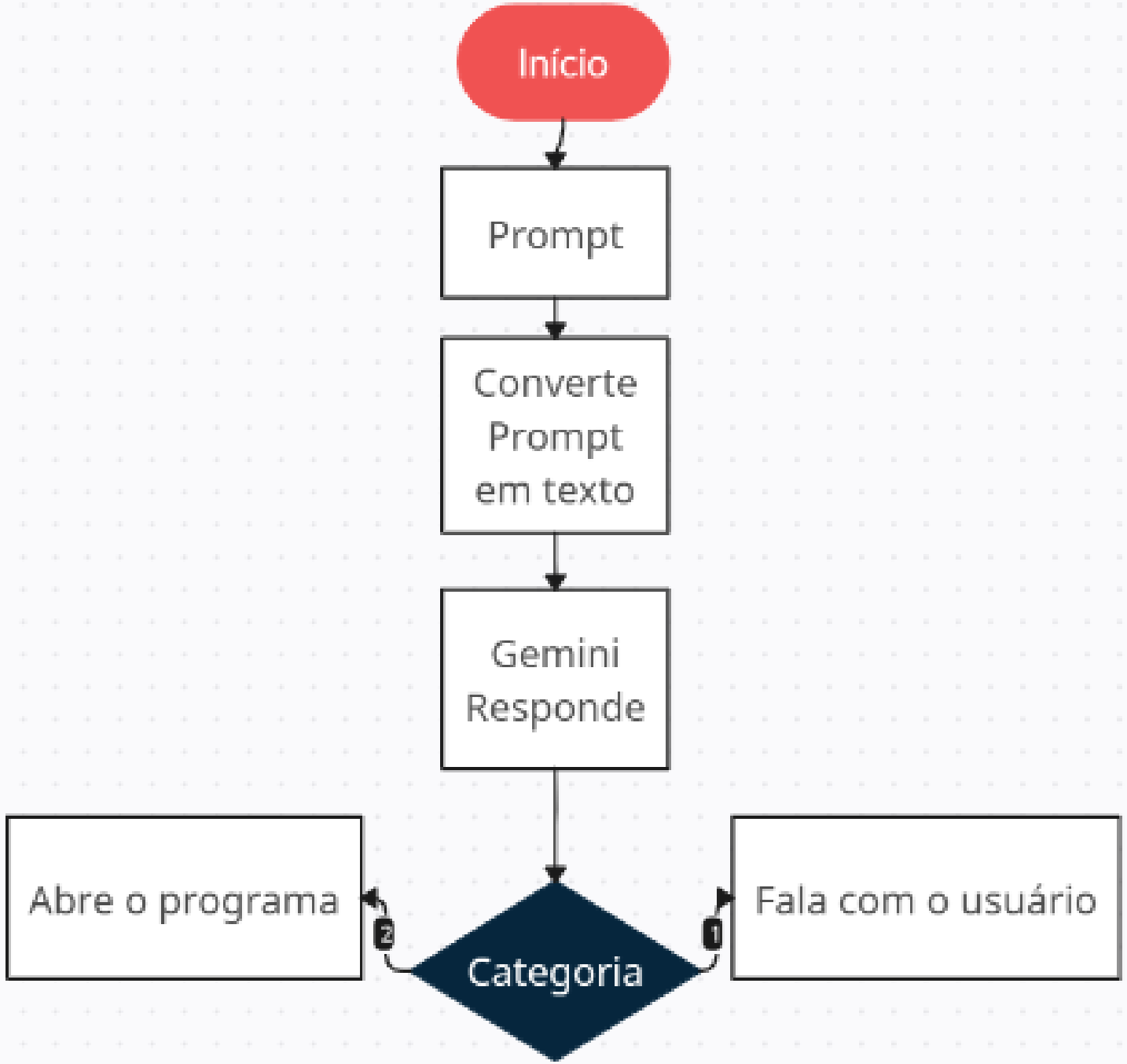
USAR O COMPUTADOR USANDO O GEMINI

User Edited

Você é um assistente para pessoas com deficiência visual, você tem 2 funções:
1- Conversar com o usuário(Descrever o que aparece na tela, ensinar o usuário a executar uma ação)
2- Executar comandos desejados pelo usuário
Você receberá um input do usuário e deve retornar um output de acordo com as duas categorias, identifique em qual categoria o input se encaixa.
Na 1 categoria você deve conversar normalmente com o usuário.
Na 2 categoria você deve dar respostas curtas seguindo o padrão apresentado. Não responda com a palavra Output:




Início
Prompt
Converte Prompt em texto
Gemini Responde
Categoria
2
1
Abre o programa
Fala com o usuário

User

Você sempre receberá uma imagem.
Na categoria 1 você deve ajudar o usuário com base na imagem enviada
Na categoria 2 você deve identificar qual elemento o usuário quer interagir e retornar exatamente o nome desse elemento.

User Edited

Exemplo categoria 2
Input: Abra o google chrome
2
Google Chrome

Exemplo categoria 2
2
Google Chrome

Exemplo categoria 2
input:Abre o edge
2
Microsoft Edge

Get code

You can call this prompt from the Gemini API by copying the following code into your project

JavaScript | Python | Android (Kotlin) | Swift | Copy

```python
import os

import google.generativeai as genai

genai.configure(api_key=os.environ["GEMINI_API_KEY"])

# Create the model
# See https://ai.google.dev/api/python/google/generativeai/GenerativeModel
generation_config = {
  "temperature": 1,
  "top_p": 0.95,
  "top_k": 64,
  "max_output_tokens": 8192,
  "response_mime_type": "text/plain",
}
```

```python
import os
import win32com.client
import google.generativeai as genai
import pyautogui
import pyttsx3
import time
import speech_recognition as sr

def audio_para_texto():
    rec = sr.Recognizer()

    texto = ''
    with sr.Microphone(1) as mic:
        rec.adjust_for_ambient_noise(mic)
        texto_para_audio('Fale o comando')
        audio = rec.listen(mic)
        texto = rec.recognize_google(audio,language='pt-BR')

    return texto
```

```python
def texto_para_audio(texto):

    speaker = pyttsx3.init()

    voices = speaker.getProperty('voices')

    speaker.setProperty('voice',voices[0].id)
    rate = speaker.getProperty('rate')
    speaker.setProperty('rate', 250)

    speaker.say(texto)
    speaker.runAndWait()
```

```python
chat_session = model.start_chat(
  history=[
    {
      "role": "user",
      "parts": [
        "Você é um assistente para pessoas com deficiência visual, você tem 2 funções:\n1- Conv
        image_drive0,
        "Você sempre receberá uma imagem.\nNa categoria 1 você deve ajudar o usuário com base n
        "Exemplo categoria 2\nInput: Abra o google chrome\n2\nGoogle Chrome\n\nExemplo categori
      ],
    },
  ]
)
```

```python
def obter_caminho_do_arquivo_do_atalho(caminho_do_atalho):
    shell = win32com.client.Dispatch("WScript.Shell")
    atalho = shell.CreateShortcut(caminho_do_atalho)
    return atalho.TargetPath
```

✓ 0.0s

```python
def encontrar_atalho_por_nome(pasta, nome_do_atalho):
    for arquivo in os.listdir(pasta):
        if arquivo.endswith(".lnk") and nome_do_atalho in arquivo:
            caminho_do_atalho = os.path.join(pasta, arquivo)
            try:
                caminho_do_arquivo = obter_caminho_do_arquivo_do_atalho(caminho_do_atalho)
                return caminho_do_arquivo
            except Exception as e:
                print(f"Erro ao processar o atalho {caminho_do_atalho}: {e}")
    return None
```

```
pergunta = audio_para_texto()
print(pergunta)
while(pergunta != 'sair'):
    chat_session.send_message(image_drive0)
    resposta = chat_session.send_message(pergunta)
    resposta = resposta.text
    print(resposta)
    if(resposta[0] == '2'):
        nome_do_atalho = resposta[2:]#Os indices anteriores são para categoria e pular linha

        pasta_dos_atalhos = r"C:\Users\Public\Desktop"
        nome_do_atalho = nome_do_atalho.strip()
        caminho_do_arquivo = encontrar_atalho_por_nome(pasta_dos_atalhos, str(nome_do_atalho))
        os.startfile(str(caminho_do_arquivo))
        time.sleep(5)
        tirar_print_da_tela('teste_gemini.png')
        image_drive0 = upload_to_gemini("teste_gemini.png", mime_type="image/png")
    else:
        texto_para_audio(resposta[2:])
    pergunta = audio_para_texto()
    print(pergunta)
```